1: modelo de ciclo de vida de software: definición.

El término **ciclo de vida del software** describe el desarrollo de software, desde la fase inicial hasta la fase final. El propósito de este programa es definir las distintas fases intermedias que se requieren para **validar** el desarrollo de la aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y **verificación** de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados.

2: identificar que es un ciclo de vida iterativo: caracteristicas básicas

**en un ciclo de vida iterativo e incremental se va liberando parte del producto (prototipos) periódicamente, en cada iteración.** En vez de una única fase final de elaboración del producto, en el ciclo de vida iterativo e incremental hay múltiples fases intermedias (tantas como iteraciones), en las que se elabora parte del producto (prototipo); cada prototipo incrementa la funcionalidad y mejora al anterior; cada iteración hace “crecer y mejorar” el producto. El producto se desarrolla por ciclos (iteraciones).

3: modelo de ciclo de vida basados en componentes: definición

El desarrollo de software basado en componentes permite reutilizar piezas de código preelaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión.

4: corba: qué es?

Common Object Request Broker Architecture). CORBA es un estándar definido por el OMG (Object Management Group) que permite escribir componentes de software en múltiples lenguajes y ejecutarlos en múltiples computadoras para trabajar juntos.

5: modelo de ciclo de vida de desarrollo rápido: definición

AD (por sus siglas en inglés) o desarrollo rápido de aplicaciones, es un modelo de diseño que se utiliza para el desarrollo de software y de sistemas de información en el menor tiempo posible. Esto se logra al centrarse en la racionalización del proceso de generación, llamado "ciclo de desarrollo". Este objetivo se consigue mediante la definición de componentes distintos. El sistema RAD se divide en cuatro fases distintas: planificación de requerimientos, diseño de casos de uso, construcción e implementación.

6: que es UML.

Unified Modeling Language - Lenguaje Unificado de Modelado). UML es un popular lenguaje de modelado de sistemas de[software](http://www.alegsa.com.ar/Dic/software.php). Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. Entre otras palabras, UML se utiliza para definir un sistema de software.

7: que es un diagrama de clases.

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

Un diagrama de clases esta compuesto por los siguientes elementos:

* [Clase](http://users.dcc.uchile.cl/~psalinas/uml/modelo.html#clase): atributos, métodos y visibilidad.
* [Relaciones](http://users.dcc.uchile.cl/~psalinas/uml/modelo.html#relacion): Herencia, Composición, Agregación, Asociación y Uso.

8: que es una clase en programacion orientada a objetos

Los datos y las operaciones comunes a un conjunto de objetos

forman un conjunto que se conoce como clase.

9: que es la herencia en programacion orientada a objetos

La herencia es la transmisión del código entre unas clases y otras. Para soportar un mecanismo de herencia tenemos dos clases: la clase padre y la/s clase/s hija/s. La clase padre es la que transmite su código a las clases hijas. En muchos lenguajes de programación se declara la herencia con la palabra "extends"

10: que es el polimorfismo en programacion orientada a objetos

polimorfismo a la capacidad que tienen los

objetos de una clase de responder al mismo mensaje o evento en función de los parámetros utilizados

durante su invocación. Un objeto polimórfico es una entidad que puede contener valores de

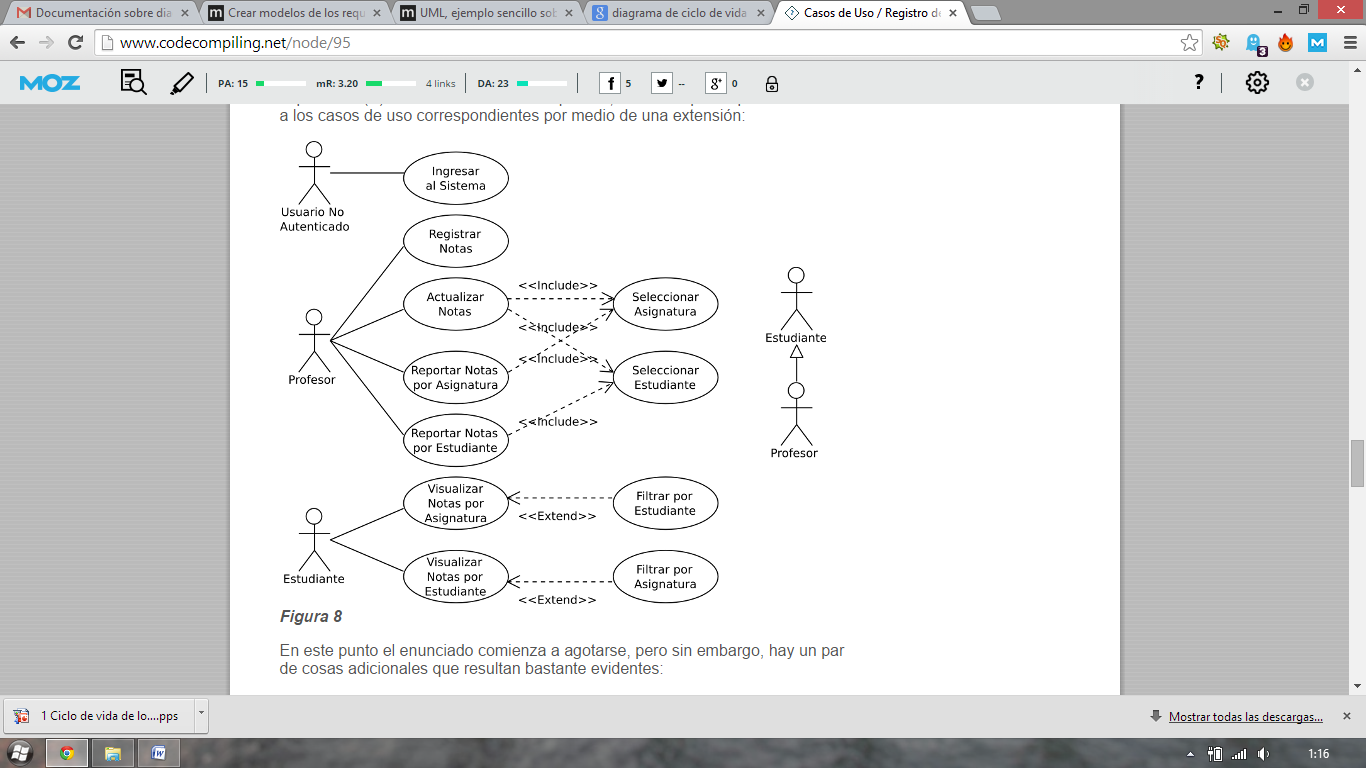
diferentes tipos durante la ejecución del programa

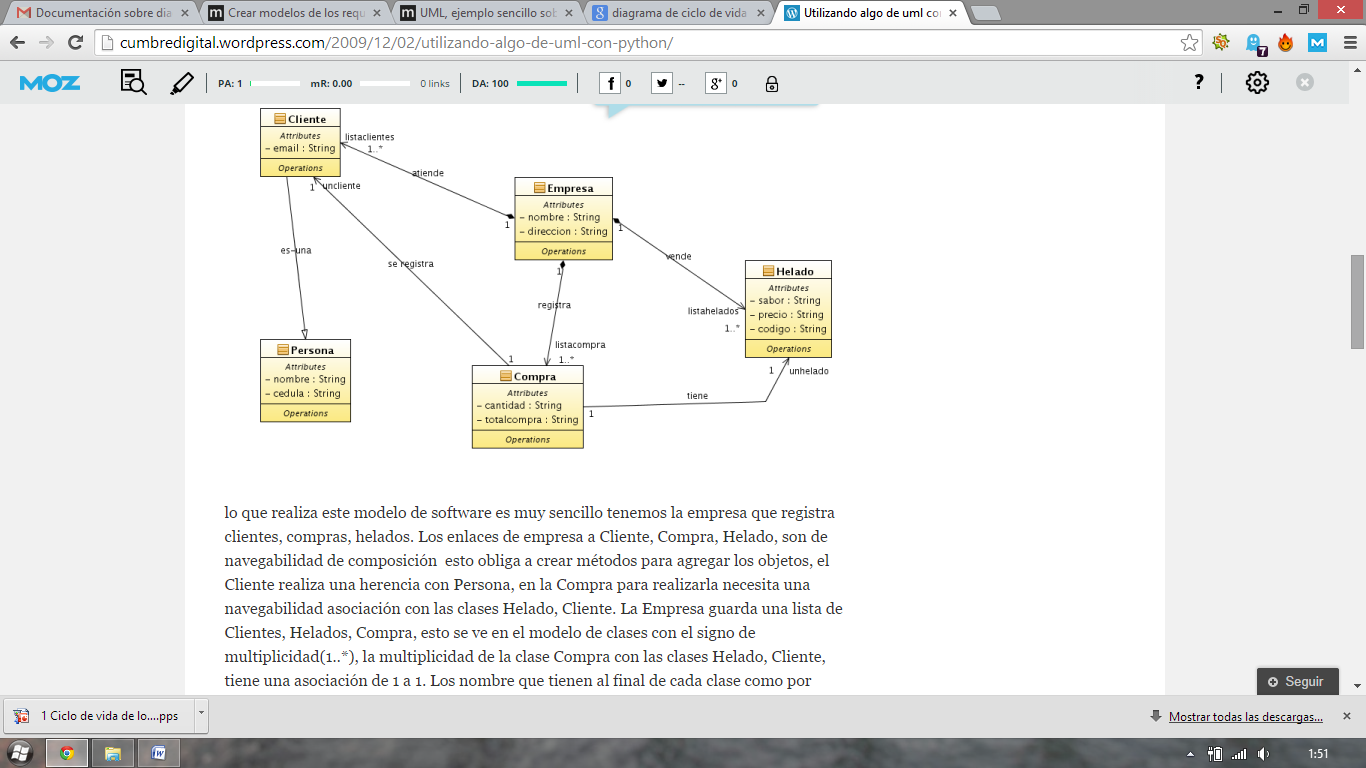
Practica: Plantear un proyecto (descriptivo, sin detallar) y elegir un modelo de ciclo de vida por el que desarrollaría este

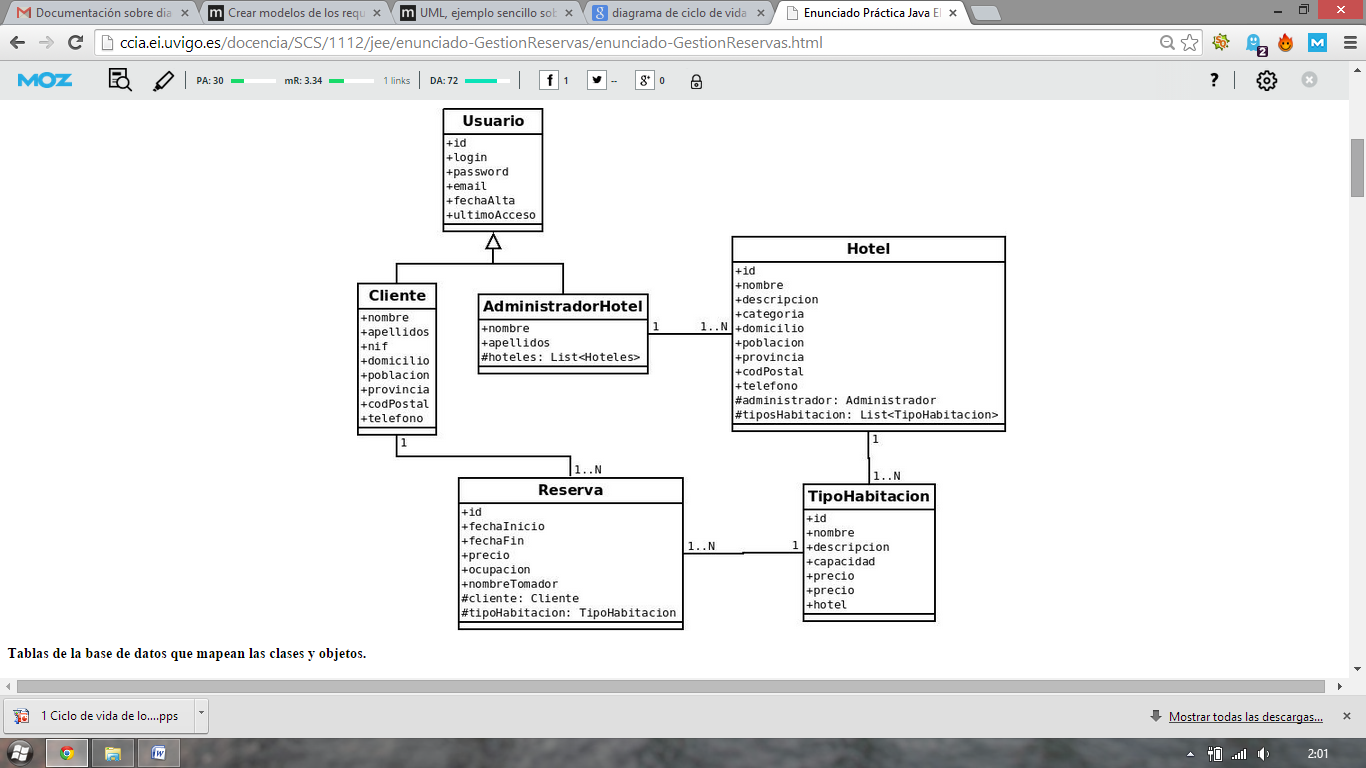
proyecto.

Hay que plasmarlo y justificar la elección.

Representar el proyecto mediante un diagrama de clases de UML.







**Capa de presentación WEB**

La aplicación WEB a desarrollar ofrecera dos conjuntos de funcionalidades:

* las destinadas a los *Clientes*: búqueda de Hoteles/Habitaciones, confección de Reservas y gestión de sus Reservas
* las destinadas a los *AdministradoresHotel*: alta y modificación de sus Hoteles, consulta y gestión de sus Reservas

El desarrollo de la capa de Presentación WEB, junto con los elementos de la capa de Negocio (EJBs) necesarios en cada caso se repartirá del siguiente modo:

**Funcionalidades para los ADMINISTRADORES de HOTELES de la central de reservas**

1. Listado de los Hoteles gestionados
2. Mantenimiento de los datos de los Hoteles y sus Tipos de Habitación
   * Altas de nuevos Hoteles (junto con sus Tipos de Habitación)
   * Modificación de Hoteles (junto con alta de nuevos Tipos de Habitación o su modificación)
3. Listado y gestión de las Reservas del Hotel actual (uno de los gestionados por este AdministradorHotel). Permitirá:
   * Ver las todas las Reservas (anteriores y actuales) del Hotel
   * Modificar Reservas aún no iniciadas (fecha de inicio posterior a fecha actual)
   * Cancelar Reservas aún no iniciadas (fecha de inicio posterior a fecha actual)

**Funcionalidades para los CLIENTES de la central de reservas**

Deberán implementar las páginas JSF (junto con los EJB [``casos de uso'' 1 a 4] que estas precisen) para dar soporte a :

1. Listado y gestión de las Reservas del Cliente actual. Permitirá:
   * Ver las todas las Reservas (anteriores y actuales) del Cliente
   * Modificar Reservas aún no iniciadas (fecha de inicio posterior a fecha actual)
   * Cancelar Reservas aún no iniciadas (fecha de inicio posterior a fecha actual)
2. Confección de una nueva Reserva, conforme a los sguientes pasos
   * Búsqueda y presentación de Hoteles (junto con sus Tipos de Habitación) en una Localidad
   * Selección de un Hotel y comprobación de Disponibilidad en base a Fecha de Entrada, Fecha de Salida y Número de Ocupantes, mostrando la lista de Tipos de Habitación disponibles
   * Selección de Tipo de Habitación, confección de la Reserva y confirmación.

**Nota:** Si se prefiere, podrá seguirse otro esquema más directo en la búsqueda, comprobación de disponibilidad y confección de la Reserva (por ejemplo: indicar directamente en la primera búsqueda la fechas de inicio y fin y el número de ocupantes)

La página JSF a partir de la cual trabajar será **gestionCliente.xhtml**